COSMO*quick* User Guide

Version 1.3

Copyright by

COSMO*logic* GmbH & Co KG

Imbacher Weg 46, 51379 Leverkusen

Germany

cosmotherm@cosmologic.de

www.cosmologic.de

Contents

1. Introduction 1
   1.1. Fragmentation Approach (COSMO*frag*) 2
   1.2. What is a SMILES string and how to get them 2
   1.3. Installation 2
   1.4. Current COSMO*quick* Limitations 2
   1.5. License 3
   1.6. Overview on Currently Predictable Properties 3
   1.7. COSMO*quick* File Menu 5
2. COSMO*quick* Tutorial 7
   2.1. Solubility Calculation and Solvent Screening with COSMO*quick* 7
   2.2. Cocrystal/Solvate Screening with COSMO*quick* 12
   2.3. Sorption & Solubility in Polymers 15
   2.4. Exporting .mcos Files 17
   2.5. COSMO*frag* Input Generator 17
   2.6. Other Available Options 20
3. Technical Details of COSMO*quick* 20
   3.1. Solubility Calculation 20
   3.2. Solubility Definitions and Unit Conversion 22
   3.3. Cocrystal Screening 22
   3.4. Solute Backfitting 23
   3.5. ADME & QSPR Calculations 24
   3.6. QSPR Builder 26
   3.7. Prediction of Hansen Solubility Parameter 27
   3.8. Generation of σ-Profiles /Fragmentation Calculation 28
   3.9. Treatment of Polymers 29
   3.10. Treatment of Charged Molecules 30
   3.11. Scripting in COSMO*quick* 30

References 32

Index 33

1. Introduction

COSMO*quick* is a graphical user interface (GUI) and a driver for COSMO*frag* [1]. The program is particularly suited for solubility calculations and screening of large data sets (e.g. cocrystal screening or partitioning coefficients). The COSMO*quick*/COSMO*frag* approach allows for quick generation of σ-profiles avoiding costly quantum chemical calculations. It relies on a database of previously computed σ-profiles for a set of about 111000 compounds (COSMO*frag* database, CFDB). Those instantenously generated σ-profiles can be used to perform COSMO*therm* like calculations with only little loss of accuracy. COSMO*quick* is a shortcut tool mainly designed for the screening of large data sets. For high quality results and accurate predictions we recommend to use COSMO*therm* together with quantum mechanically derived σ-profiles. COSMO*therm* is a full implementation of COSMO-RS theory and is also distributed by COSMO*logic*.

Currently the following calculation modes can be carried out with COSMO*quick*:

- Prediction of solubilities with multiple reference solvents and relative solubilities [3.1]
- Cocrystal screening, i.e. fast calculation of excess enthalpies [3.3]
- Prediction of the sorption of small molecules in polymers or solvents [2.3 & 3.9]
- Creation of the sigma-profile of a unknown/undetermined compound (could be anything) by using reference solubilities in several solvents. [3.4]
- ADME properties calculations, i.e. different partition coefficients & water solubility [3.5]
- QSPR calculations using multi-linear regression or random forest based models [3.5]
- Generation and deployment of QSPR models using COSMOquick derived descriptors [3.6]
- Generation of Hansen solubility parameters via solubility prediction [3.7]
- Generation of approximate σ-profiles for COSMO*therm* calculations [3.8]

COSMO*quick* and COSMO*frag* are based on COSMO-RS theory, which has become an efficient and versatile tool for the prediction of a large variety of physicochemical properties, especially in its efficient implementation within the COSMO*therm* program. Based on quantum chemical (DFT/COSMO) calculations for the individual molecules it allows for physically most sound estimations of general vapour-liquid and liquid-liquid equilibria and of related properties like solubilities and partition coefficients. In addition it has been extended to properties like drug- and pesticide solubility, blood-brain partition coefficients, intestinal absorption, soil sorption coefficients, etc. which are of importance in the design and development of drugs, pesticides and other physiological agents. For more information on the COSMO*therm* program suite please contact info@cosmologic.de.

All publications resulting from use of this program must acknowledge the following:

C. Loschen, A. Hellweg, A. Klamt, COSMO*quick*, Version 1.3; COSMOlogic GmbH & Co. KG, Leverkusen, Germany, 2014.

In Addition reference 8 should be cited.

1.1. Fragmentation Approach (COSMO*frag*)

COSMO*quick* internally calls COSMO*frag* for the generation of σ-profiles and for the calculation of properties, detailed information on COSMO*frag* can be found in Reference 1. The basic idea for the fragmentation approach is the composition of the σ-profile of a new molecule from existing σ-profiles of molecules that have already been pre-calculated. Currently there are more than 111.000 diverse molecules stored within the CFDB. Thus, there is no need for quantum chemical calculations prior to COSMO-RS calculations of a new molecule. The drawback is a little loss of accuracy for molecules which are composed from several fragments from the CFDB. If a new molecule is fragmented into a lot of CFDB molecules it may be badly represented. Therefore, the number and quality of the fragments used for a fragmentation (i.e. σ-profile generation) calculation should be monitored (see section 3.8).

1.2. What is a SMILES string and how to get them

COSMO*quick* relies to a large extent on SMILES strings, which are used as molecular input for any calculations. SMILES stands for **Simplified Molecular Input Line Entry Specification**. It allows for the descriptions of the structure of molecules using comparatively short ASCII codes. Examples for some simple compounds are: Propane: CCC, Ethanol: CCO, oxalic acid: C(C(=O)O)(=O)O. Within COSMO*quick* they may be obtained with the 2D structure editor which automatically creates a SMILES string for the user or via the web-service which can be found under TOOLS in the menu. Molecules encoded in the InChi (**IUPAC International Chemical Identifier**) format can be loaded with the 2D structure editor which will convert them into a SMILES string. Additionaly SDF files may be used as input for COSMO*quick*.

1.3. Installation

COSMO*quick* is shipped with an installer for Windows, Linux and MacOS. The COSMO*frag* database CFDB needs to be installed separately. Extract the COSMO*frag* database CFDB.zip to a folder of your choice. Please note, that you need an actual unzipping program (e.g. 7-zip), some older versions of Winzip may cause problems here. Furthermore, due to the size of the database of about 2.4 GB the unzipping process may take several minutes. All subdirectories are automatically created. At the first start-up of the software you are asked to specify the location of the CFDB. Please choose an appropriate directory. Access to the CFDB over the network may slow down the fragmentation significantly.

Proxy-Server: Using the NIH web-service needs direct access to the internet. In case you want to use this service and you have to access the internet via a proxy-server you will have to adapt the java configuration file "COSMOquick.vmoptions" which can be found in the COSMOquick subdirectory in the installation directory. Simply umcomment the respective line there and use your companies/institutions proxy settings.

1.4. Current COSMO*quick* Limitations

The COSMO*quick* approach to generate approximate σ-profiles leads to certain limitations in the application of the method:

- No conformer treatment is possible with COSMO*quick.*
- For most common *ionic* compounds σ-profile can be generated with COSMO*quick*, but property prediction is currently not recommended.

- A few complex drugs may not be properly represented in the COSMO*quick* database and no valid σ-profile may be generated. (For those cases an Error/Warning message is shown.) For those cases .cosmo files have to be generated and added to the database.
- Known SMILES issues are: Implicid H inside square brackets is not supported, e.g. write C or [NH4+] instead of [C] or [N+].
- COSMO*quick* has been tested to run with 20000 medium sized organic compounds. Higher numbers may be feasible with the GUI but for performance reasons for large sets of compounds we recommend to use the command-line based COSMO*frag* instead. Input files for COSMO*frag* may be created, loaded or modified via a graphical user interface from TOOLS->COSMO*frag* calculation.
- There is currently the restriction to use a parameterization at the BP-SVP-COSMO level
- For larger set of compounds make sure that sufficient disk space is available. A computation of 10000 compounds needs currently roughly 500M for temporary data. In case the GUI rans out of memory additional memory can be allocated via changing the -Xmx1024m options in the COSMOquick.vmoptions file in the COSMOquick directory.
- Length of input SMILES is limited to a total number 222 atoms.

Limitations due to third party software used within COSMO*quick*:

- Limited support for inorganic compound SMILES.
- JChempaint (2D structure editor) may display some compounds incorrectly, like cis/trans isomers
- The NIH webservice Chemical Identifier Resolver is in the public domain and a proper continous functioning can not be guaranteed by us.

1.5. License

Currently the license is checked via COSMO*frag* which is called internally by COSMO*quick*. Please provide a valid license file at the first startup of the software. Please note that the COSMO*frag* executable shipped with COSMO*quick* is only able to use parameterization at the BP-SVP-COSMO level. For higher level calculations we recommend to use COSMO*therm* instead.

1.6. Overview on Currently Predictable Properties

COSMO*quick* predicts several thermodynamic properties; the following table summarizes those properties and lists where they can be found:

Property	Quantity	Module
Solubility	log10(x), x in mole fraction S in mol/L S in g/L w in g/g	Solubility Prediction
Free energy of fusion	ΔG_{fus} in kcal/mol	Solubility Prediction, as computed from experimental solubilities
Free energy of fusion	ΔG_{fus} in kcal/mol	QSPR & ADME, as QSPR estimate
Activity coefficient	lnγ	Solubility Prediction, Henry constant & gas solubility
Excess enthalpy of Compound A and B	H_{ex} in kcal/mol	Cocrystal and Solvate Screening
Free energy of mixing of A and B	ΔG_{mix} in kcal/mol	Cocrystal and Solvate Screening
Henry constant	H in bar	Henry constant & gas solubility
Vapor pressure	p(vapor)	Henry constant & gas solubility
Free energy of solvation	ΔG_{solv} In kcal/mol	Henry constant & gas solubility
Gas solubility	S in cm^3/(cm^3 bar)	Henry constant & gas solubility
Melting point	Tm, K	QSPR & ADME
Enthaly of fusion	ΔH_{fus} in kcal/mol	QSPR & ADME
Water solubility	logS(water) S in mol/L w in g/g log10(x), x in mole fraction	QSPR & ADME, Solubility Prediction
Octanol-water partitioning coefficient	logKow	QSPR & ADME
Blood-Brain partitioning coefficient	logBB	QSPR & ADME
Plasma-protein (Human Serum Albumin) partitioning.	logKHSA	QSPR & ADME
Intestinal Absorption coefficient	logKIA	QSPR & ADME
Organic carbon (Soil)-Water partition coefficient	logKOC	QSPR & ADME
Abrahams parameter	E,S,A,B,V	QSPR & AMDE
Hansen parameter	δD,δP,δH	Hansen parameter estimation

1.7. COSMO*quick* File Menu

The following options are available in the COSMO*quick* file menu:

FILE:

NEW JOB: Starts a new job and closes all results windows.

LOAD: Either load a file containing SMILES strings and compound names (".smi") or a previous fragmentation run (".frg").

QUICKLOAD: Loads the last fragmentation run.

OPEN TEMPORARY DIRECTORY: Opens the temporary directory used for calculations.

EXTRAS:

GLOBAL OPTIONS: Options for COSMO*frag* and (internal) COSMO*therm* runs can be set here.

GENERAL SETTINGS: Here you can specify for example the location of the COSMO*frag* executable, the COSMO*frag* database (CFDB) and the license file.

SHOW LOG: Opens a log window with additional information on what is currently happening, i.e. it basically makes the standard output (stdout) available.

TOOLS:

CREATE NEW QSPR MODEL: Build a QSPR model via linear regression based on the available COSMO*quick* descriptors.

COSMOFRAG CALCULATION: A user interface for starting individual COSMO*frag* jobs, COSMO*sim* jobs and loading and saving COSMO*frag* input files. This allows for additional flexibility as compared to the standard COSMO*quick* workflow.

REQUEST SMILES: This allows for retrieving SMILES string from a NIH webservice (CIR – chemical resolver identifier). Please note that this web service is under public domain and no guaranty can be provided for its correct functionality.

SOLUBILITY CONVERTER: This tool allows for a conversion between the different definitions of solubility which can be found in the literature.

CREATE .FCOS FILES: Create approximate 3D .cosmo files (.fcos) from .xyz or .sdf input files.

AUTOMATICALLY CREATE 3D STRUCTURES: Use the UFF or the MMFF94 forcefields to create 3D structures from SMILES.

LICENSE:

IMPORT LICENSE: Use this button to import a new license file (license.ctd) into the program.

HELP:

COSMO*quick* USER GUIDE: Opens the COSMO*quick* manual as pdf documents.

COSMO*frag* REFERENCE MANUAL: Opens the COSMO*frag* manual as pdf documents.

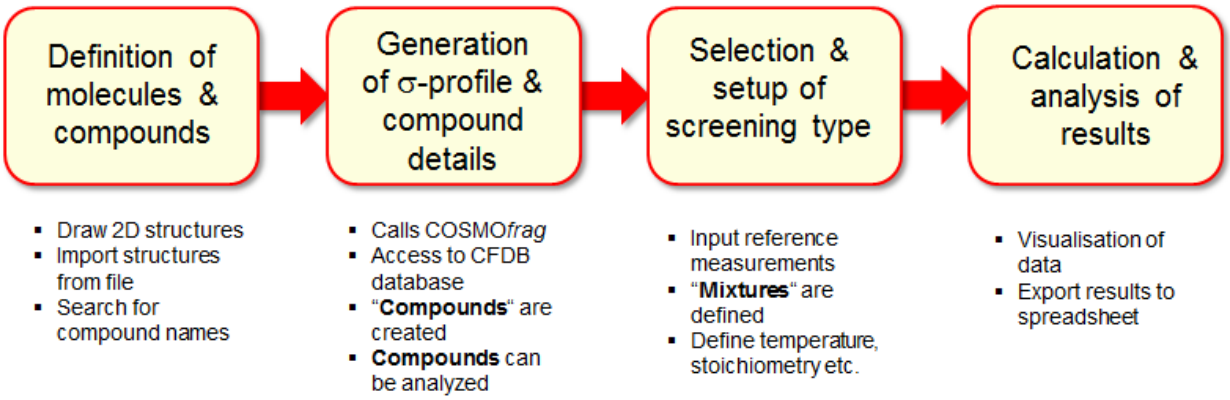
ONLINE SOURCES: Watch online introduction into COSMO*quick*

ABOUT COSMOQUICK: Gives information on COSMO*quick* and also about the current used license.

LICENSE AGREEMENTS USED: Shows all currently used external licenses of COSMO*quick*.

2. COSMO*quick* Tutorial

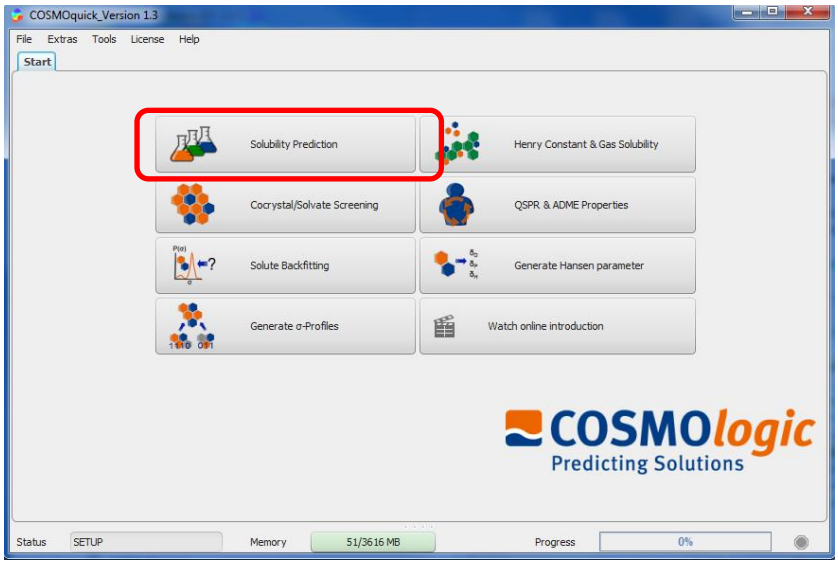
Before starting with a specific tutorial it is helpful to have a look at the typical COSMO*quick* workflow:



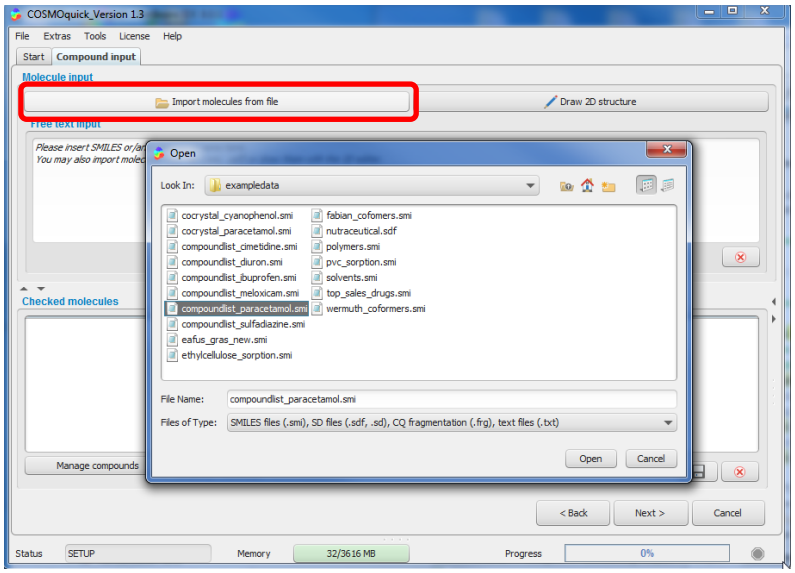
The first step consists of defining the molecules under scrutiny, this is usually done by loading a file, drawing a structure, or defining a SMILES. Afterwards the compounds are being analyzed and the database (CFDB) is accessed for the generation of the COSMO-RS σ-profiles. Then usually the type of calculation is specified and specific parameters (stoichiometry, temperature) can be chosen. Then, in most cases a COSMO*therm* calculation is being done internally based on the σ-profiles generated before and results are presented in tabulated and in graphical form.

2.1. Solubility Calculation and Solvent Screening with COSMO*quick*

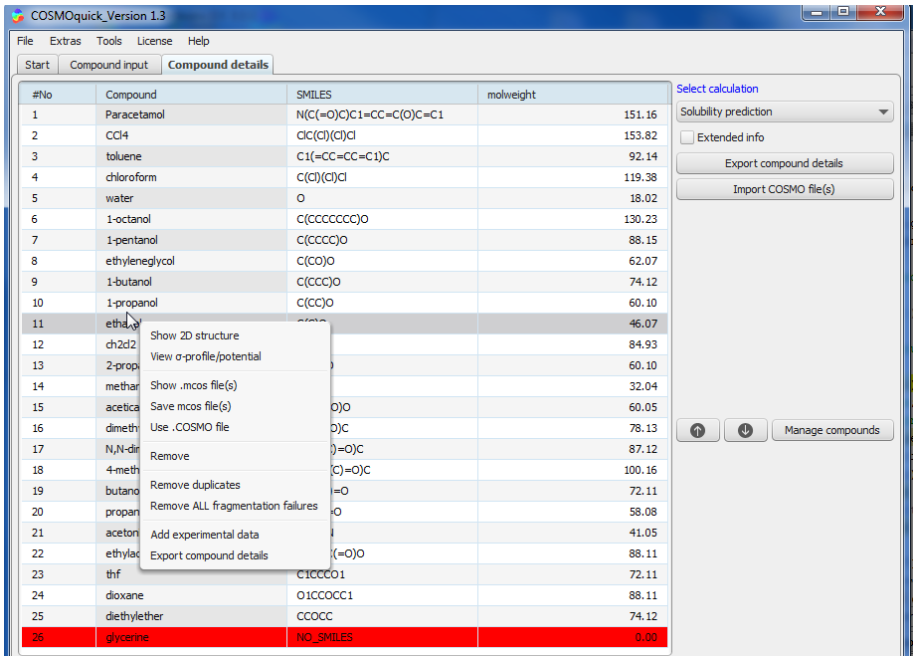
This section describes how to perform a COSMO*quick* solubility calculation with reference solubilities. Please have a look at chapter 3.1 for details of the procedure. After the first startup please provide a location for the COSMO*frag* database (CFDB) and also for a valid license file. If the CFDB location and the license are OK, you arrive at the start screen and may choose the calculation type; please choose "Solubility Prediction":



Now you arrive at the compound setup, where you can specify the molecules you want to study. Please select “Import molecules from file” and open the .smi file compoundlist_paracetamol.smi from the directory “exampledata”.

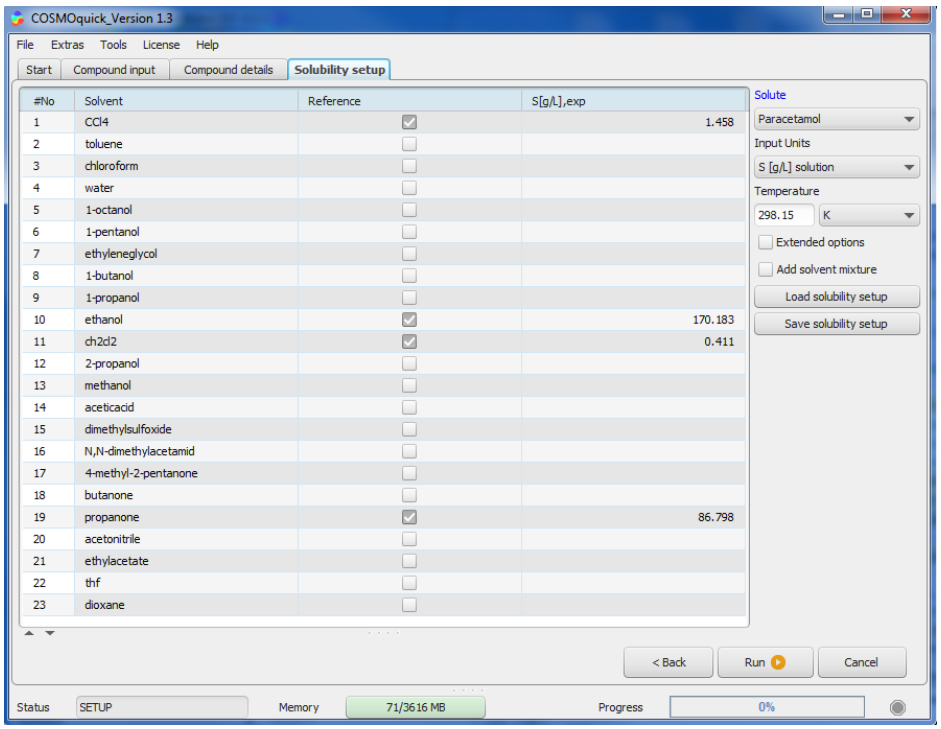


You will now find a list of SMILES strings and compound names in the lower area of the compound input. You can add a compound by adding a new line in the text area and type a name or a SMILES string. For example type “diethylether” and “glycerine” there. In the case of glycerine no SMILES is found in the internal database and the entry is marked red. If you are connected to the web, the button “manage compounds” allows you to use a web-service to look up the SMILES automatically. You may also add a compound by drawing it with the 2D structure editor. The editor will automatically generate a SMILES string for you which you can add to the compound setup. After you have created a suitable list of molecules select the “next button” at the bottom. Now a fragmentation is initiated and the CFDB is being accessed which may take a while. After it is finished the screen should look like:



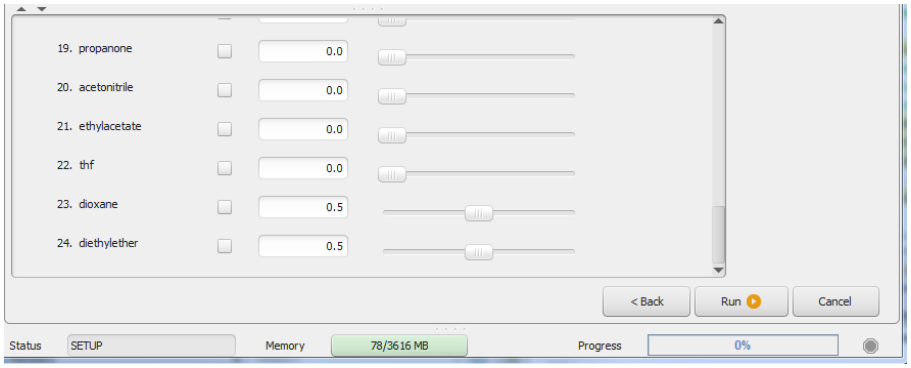
Compounds where the fragmentation has failed are marked red as in this case glycerine. This may have several reasons: The compound name was not found within the delivered database and therefore no valid SMILES was found, or a SMILES was provided but contains an element which is not available in the CFDB. The checkbox “Extended info” may reveal the reason for a failed fragmentation. In this case the name “glycerine” was just not found in the delivered database. Therefore we have to provide a SMILES string for this compound in the “Compound input” screen. This could be done either by using the “Manage compounds” button at the right or by selecting the right row and calling the context menu by a right mouse button click. In this tutorial we just remove the compound by either selecting “Remove” or “Remove ALL fragmentation failures”.

We now proceed to the next tab, where we have to select the reference solubilities and to specify experimental values for those. Paracetamol is now automatically selected as solute as it was the first molecule in the list. Please select “Load solubility setup” and choose the file “paracetamol_pure.mix” from the “exampledata” directory. The window should look like:



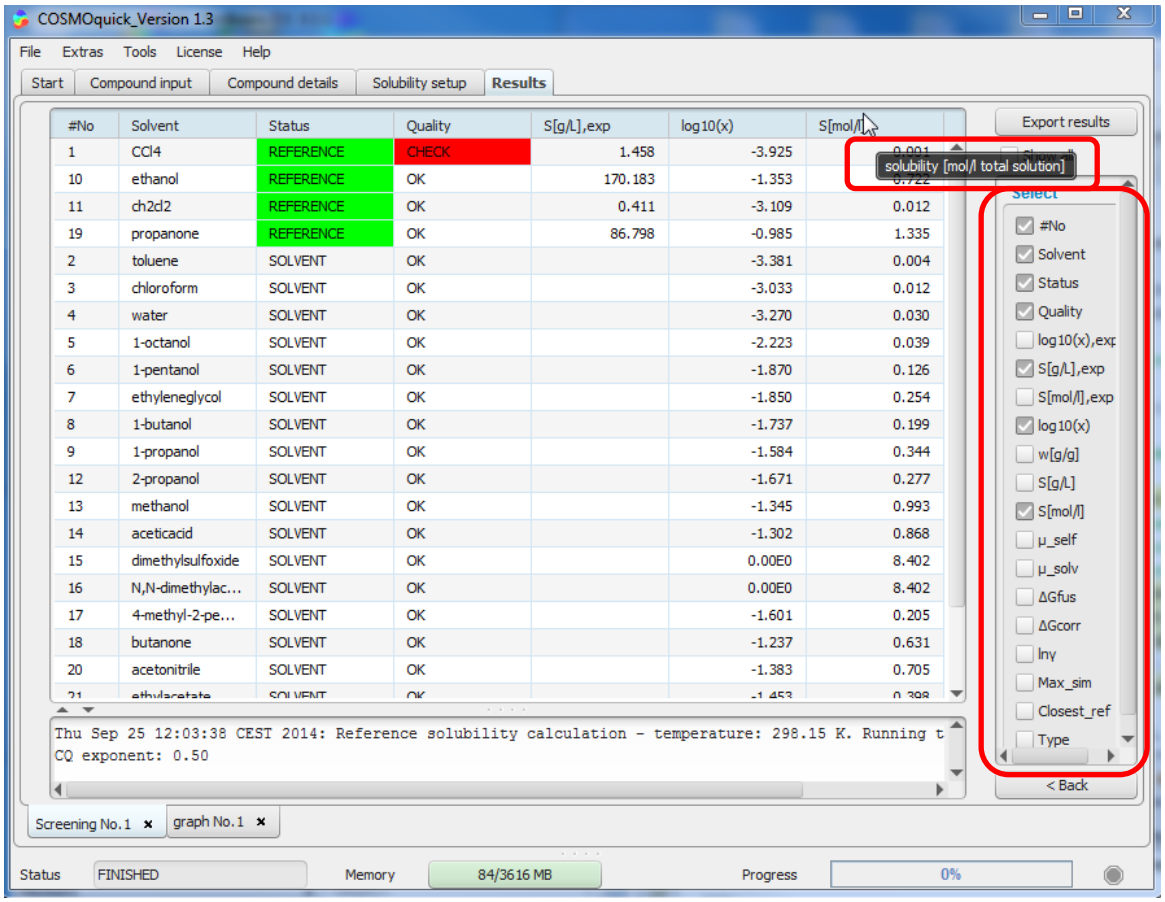
We have just loaded an experimental setup from the publication: Granberg, R. A. & Rasmuson, Å. C. Solubility of Paracetamol in Pure Solvents *Journal of Chemical & Engineering Data,* **1999***, 44*, 1391-1395. Four solvents are marked now as references: CCl4, ethanol, dichloromethane and propanone. This means that their respective solubilities are used to improve the computed solubility of similar solvents. Please note that you may specify additional solubilities for the other solvents, but only solvents which are marked are considered as references. If you do not specify any reference then a relative solubility is carried out, where all results are related to the solvent which shows the highest solubility. Please remind that this quantity is not an absolute value and may only be used to compare relative solubilities.

To add a solvent to this experimental setup you have to select the checkbox “Add Solvent mixture”. There will be now an additional area visible where you can select a compound (or several compounds), choose the composition in mole or mass fraction and specify an experimental solubility in case there is one.



You may scroll down and choose e.g. a 50:50 mixture (mole fraction) from diethyl ether and dioxane as additional solvent. Scroll up and click “Add solvent” to add this mixture to your solvent list.

After you have finished your input you may proceed and select the “Run” button which starts the solubility calculation. The calculation may take a few seconds; afterwards you find some new tabs at the bottom of the window with the results of the calculation, a table and a plot window:


COSMOquick_Version 1.3
File Extras Tools License Help
Start Compound input Compound details Solubility setup Results
Export results
solubility [mol/l total solution]
Thu Sep 25 12:03:38 CEST 2014: Reference solubility calculation - temperature: 298.15 K. Running t
CQ exponent: 0.50
< Back
Screening No.1
graph No.1
Status FINISHED
Memory 84/3616 MB
Progress 0%

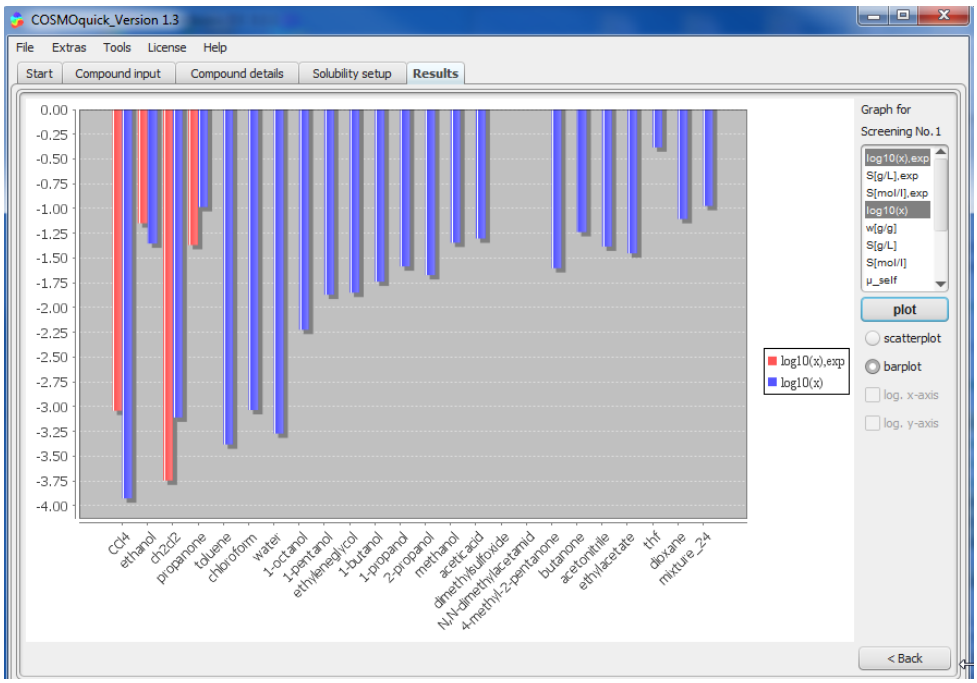
#No	Solvent	Status	Quality	S[g/L],exp	log10(x)	S[mol/l]
1	CCl4	REFERENCE	CHECK	1.458	-3.925	
10	ethanol	REFERENCE	OK	170.183	-1.353	
11	ch2cl2	REFERENCE	OK	0.411	-3.109	0.012
19	propanone	REFERENCE	OK	86.798	-0.985	1.335
2	toluene	SOLVENT	OK		-3.381	0.004
3	chloroform	SOLVENT	OK		-3.033	0.012
4	water	SOLVENT	OK		-3.270	0.030
5	1-octanol	SOLVENT	OK		-2.223	0.039
6	1-pentanol	SOLVENT	OK		-1.870	0.126
7	ethyleneglycol	SOLVENT	OK		-1.850	0.254
8	1-butanol	SOLVENT	OK		-1.737	0.199
9	1-propanol	SOLVENT	OK		-1.584	0.344
12	2-propanol	SOLVENT	OK		-1.671	0.277
13	methanol	SOLVENT	OK		-1.345	0.993
14	aceticacid	SOLVENT	OK		-1.302	0.868
15	dimethylsulfoxide	SOLVENT	OK		0.00E0	8.402
16	N,N-dimethylac...	SOLVENT	OK		0.00E0	8.402
17	4-methyl-2-pe...	SOLVENT	OK		-1.601	0.205
18	butanone	SOLVENT	OK		-1.237	0.631
20	acetonitrile	SOLVENT	OK		-1.383	0.705

You find also a red mark for row of CCl4, which means that the computed correction for this reference is significantly larger than one would expect (the threshold is currently set at 1.5 kcal/mol). A large correction term is a strong hint that this experimental value is inaccurate and should be checked. Indeed, as a personal communication from the authors of this experiment confirmed the experimental value of log10(x)=-3.04 is most probably much too high and the true

solubility of paracetamol in CCl4 is about log10(x)=-5. Please have a look at a more detailed discussion of this issue in reference 8.

You find a lot of useful additional information on the calculation by selection of the corresponding field at the right column. For example if you inspect the last column of this view you find that each solvent has assigned a type, according to its similarity with some standard solvents. The three letter codes represent the following solvent types: NONP, nonpolar (e.g. hexane), ACC, acceptor (e.g. acetonitrile), DON, donor (e.g. chloroform) and D-A, donor-acceptor (e.g. water). To cover the potential solvent space broadly and to get a good predictivity it is recommended to include one of each type as a reference, at least you should have an unpolar, an acceptor and a donor-acceptor solvent. Please note that by dragging the mouse over the field of interest you obtain some additional information (Tooltip) on that variable.

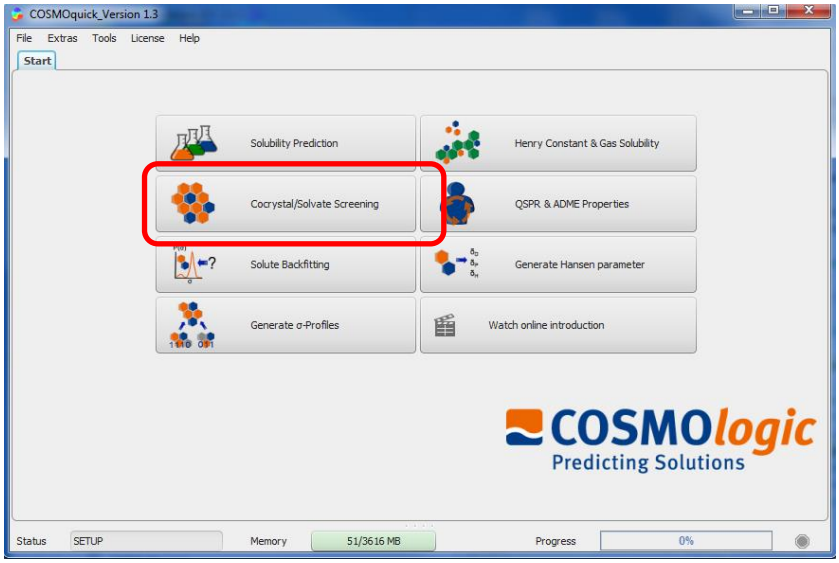
There is a second window available with plots of the computed solubilities. If you have specified experimental solubilities they are also plotted.



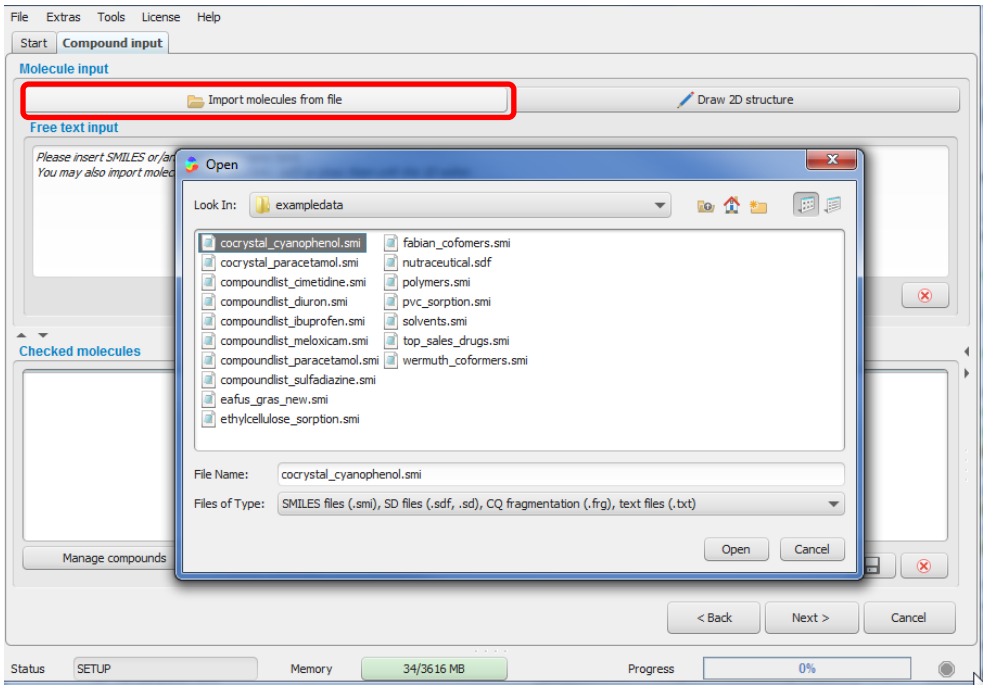
You may now extract the results either by using copy&paste on the tables (Ctrl+C/Ctrl+V) or use the export to excel/.csv function.

2.2. Cocrystal/Solvate Screening with COSMO*quick*

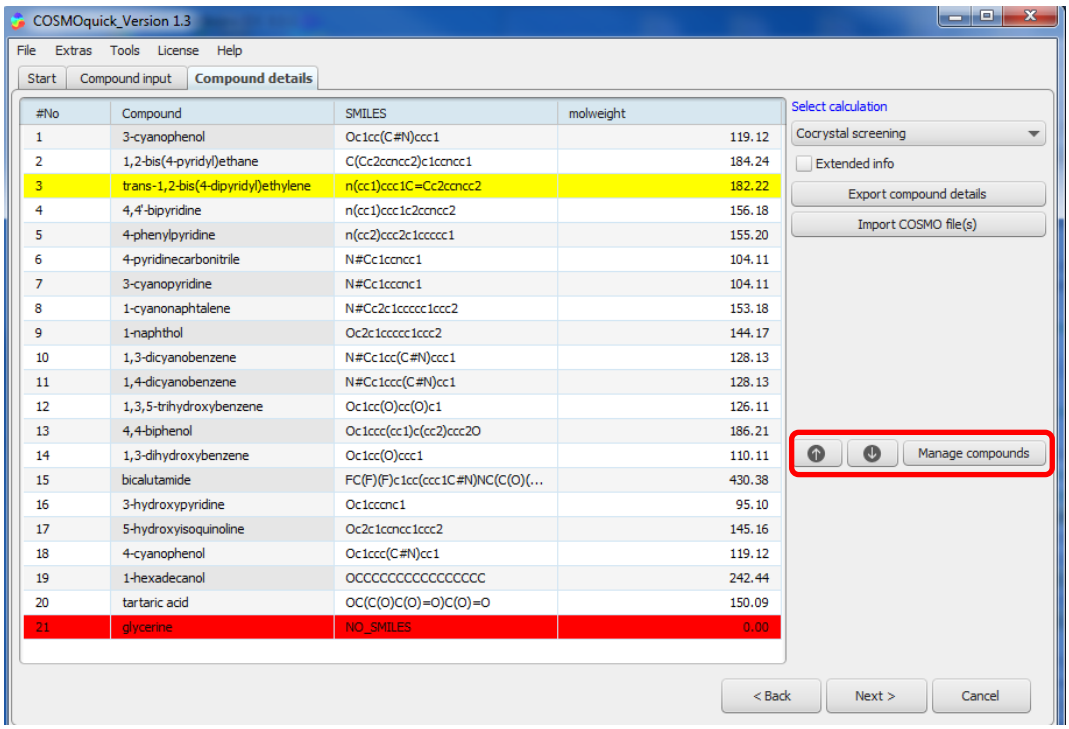
This section explains how to carry out a screening for potential coformers which can form a cocrystal with a molecule, typically an active pharmaceutical ingredient (API). This workflow can also be used to identify possible solvate forming solvents for the specific drug. Please have a look at section 3.3 for details of the procedure. Please select “Cocrystal/Solvate Screening” from the start window.



Now you arrive at the compound setup, where you can specify the molecules you want to study. Please select “Import molecules from file” and open the .smi file cocrystal_cyanophenol.smi from the directory “exampledata”.



You will now find a list of SMILES strings and compound names in the lower area of the compound setup screen. You can add a compound by adding a new line and type a name or a SMILES string in the text area above. For example type “tartaric acid” and “glycerine” there. You may also add a compound by drawing it with the 2D structure editor. The editor will automatically generate a SMILES string for you which you can add to the compound setup. After you have created a suitable list of molecules select the “Next” button at the bottom. Now a fragmentation is initiated and the CFDB is being accessed which may take a while. After it is finished the screen should look like:

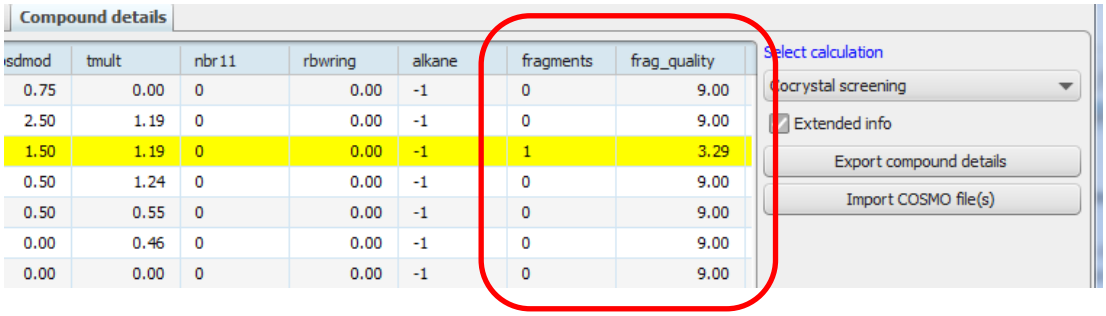


#No	Compound	SMILES	molweight
1	3-cyanophenol	Oc1cc(C#N)ccc1	119.12
2	1,2-bis(4-pyridyl)ethane	C(Cc2ccncc2)c1ccncc1	184.24
3	trans-1,2-bis(4-dipyridyl)ethylene	n(cc1)ccc1C=Cc2ccncc2	182.22
4	4,4'-bipyridine	n(cc1)ccc1c2ccncc2	156.18
5	4-phenylpyridine	n(cc2)ccc2c1ccccc1	155.20
6	4-pyridinecarbonitrile	N#Cc1ccncc1	104.11
7	3-cyanopyridine	N#Cc1cccnc1	104.11
8	1-cyanonaphtalene	N#Cc2c1ccccc1ccc2	153.18
9	1-naphthol	Oc2c1ccccc1ccc2	144.17
10	1,3-dicyanobenzene	N#Cc1cc(C#N)ccc1	128.13
11	1,4-dicyanobenzene	N#Cc1ccc(C#N)cc1	128.13
12	1,3,5-trihydroxybenzene	Oc1cc(O)cc(O)c1	126.11
13	4,4-biphenol	Oc1ccc(cc1)c(cc2)ccc2O	186.21
14	1,3-dihydroxybenzene	Oc1cc(O)ccc1	110.11
15	bicalutamide	FC(F)(F)c1cc(ccc1C#N)NC(C(O)(...	430.38
16	3-hydroxypyridine	Oc1cccnc1	95.10
17	5-hydroxyisoquinoline	Oc2c1ccncc1ccc2	145.16
18	4-cyanophenol	Oc1ccc(C#N)cc1	119.12
19	1-hexadecanol	OCCCCCCCCCCCCCCCC	242.44
20	tartaric acid	OC(C(O)C(O)=O)C(O)=O	150.09
21	glycerine	NO_SMILES	0.00

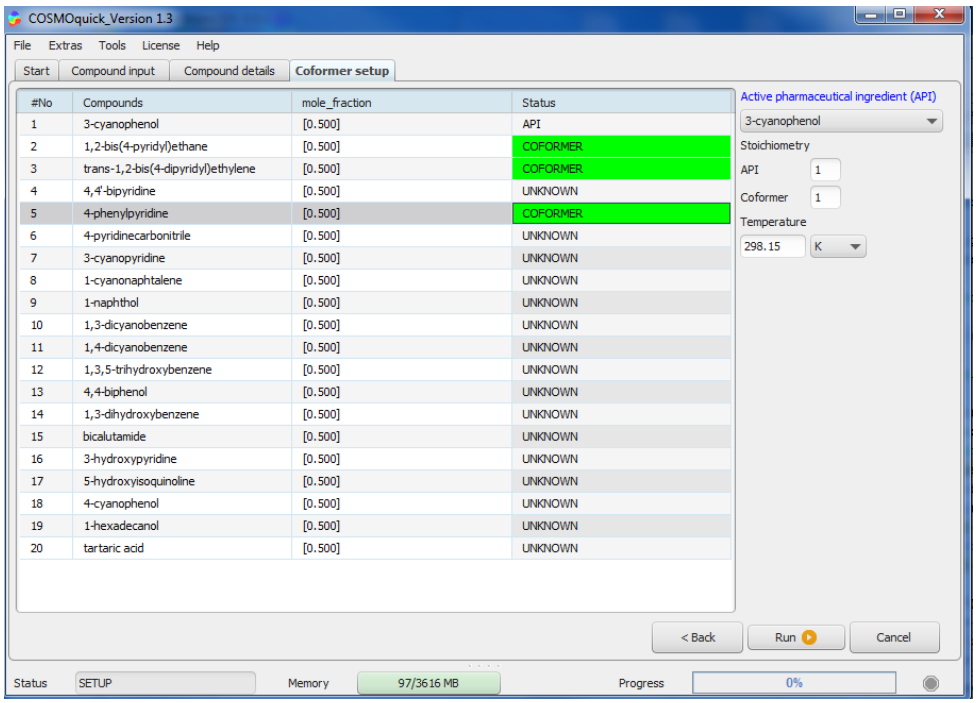
Compounds where the fragmentation has failed are marked red as in this case glycerine. This may have several reasons: The compound name was not found within the delivered database and therefore no valid SMILES was found, or a SMILES was provided but contains an atomic environment which is not available in the CFDB. The checkbox “Extended info” may reveal the reason for a failed fragmentation. In this case the name “glycerine” was just not found in the delivered database. Therefore we would have to provide a SMILES string for this compound by ourself in the Compund input screen. This could be done either by using the “Manage compounds” button or by selecting the right row and calling the context menu by a right mouse button click. Now we just remove the compound by either selecting “Remove” or “Remove ALL fragmentation failures”. The context menu may also used to specify a .cosmo file for the compound, to show the structure, the σ-profile/σ-potential, to remove duplicates etc.

The quality of a fragmentation can be assessed by the column “fragments” which becomes visible if the checkbox “Extended info” is selected. Here the number of fragments which had to be used to generate the according σ-profile for a molecule is displayed. A large number of fragments is a hint that no similar molecule is available in the CFDB. For a good cocrystal screening the number of fragments for the API itself should not be too large, otherwise the results may not be accurate. Another indicator for the quality of the fragmentation is the column labeled “frag_quality”. It contains the average similarity of each atom of the molecule with a similar environment from an entry of the CFDB, ranging from 0 (no similarity) to 9 (identity). Low values indicate a bad fragmentation and those compounds may be considered only with care for

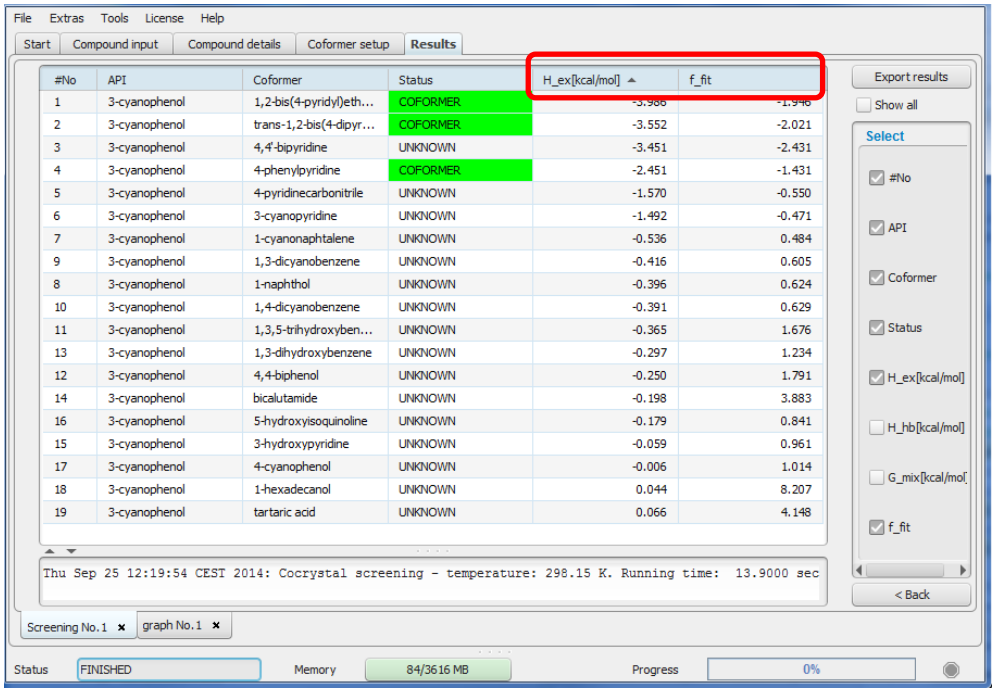
further calculations. A similarity=9 means that the compounds have been taken in a 1:1 fashion out of the database.



We now procceed to the next window where all of our compounds are listed and where one can set the API, temperature and the stoichiometry of the system under scrutiny. For unknown systems it is recommended to keep the 1:1 stoichiometry, as most cocrystals crystallize in either a 1:1 or a 2:1 ratio, where the latter would not significantly change the results within the given frame of accuracy. If we have experimental knowledge about an API-coformer system we may also select a pair as being either a cocrystal or no cocrystal by using the left mouse over the specific table entry in the status column. This just results in a coloring of the entry which may be useful if we screen a large list of compounds:



If we have a compound set of our choice (this cocrystal setup is taken from Bis et al. *Mol Pharm* **2007,** 4, 401.) we proceed by pressing the "Run" button at the lower left corner and the screening starts. After a few seconds the results of the calculation are represented in the next window. To order the API-coformer pairs according to their highest propensity of forming a cocrystal we select the column showing the excess enthalpy "H_ex" and sort it.


File Extras Tools License Help
Start Compound input Compound details Coformer setup Results
#No API Coformer Status H_ex[kcal/mol] f_fit
1 3-cyanophenol 1,2-bis(4-pyridyl)eth... COFORMER -3.988 -1.946
2 3-cyanophenol trans-1,2-bis(4-dipyr... COFORMER -3.552 -2.021
3 3-cyanophenol 4,4'-bipyridine UNKNOWN -3.451 -2.431
4 3-cyanophenol 4-phenylpyridine COFORMER -2.451 -1.431
5 3-cyanophenol 4-pyridinecarbonitrile UNKNOWN -1.570 -0.550
6 3-cyanophenol 3-cyanopyridine UNKNOWN -1.492 -0.471
7 3-cyanophenol 1-cyanonaphtalene UNKNOWN -0.536 0.484
9 3-cyanophenol 1,3-dicyanobenzene UNKNOWN -0.416 0.605
8 3-cyanophenol 1-naphthol UNKNOWN -0.396 0.624
10 3-cyanophenol 1,4-dicyanobenzene UNKNOWN -0.391 0.629
11 3-cyanophenol 1,3,5-trihydroxyben... UNKNOWN -0.365 1.676
13 3-cyanophenol 1,3-dihydroxybenzene UNKNOWN -0.297 1.234
12 3-cyanophenol 4,4-biphenol UNKNOWN -0.250 1.791
14 3-cyanophenol bicalutamide UNKNOWN -0.198 3.883
16 3-cyanophenol 5-hydroxyisoquinoline UNKNOWN -0.179 0.841
15 3-cyanophenol 3-hydroxypyridine UNKNOWN -0.059 0.961
17 3-cyanophenol 4-cyanophenol UNKNOWN -0.006 1.014
18 3-cyanophenol 1-hexadecanol UNKNOWN 0.044 8.207
19 3-cyanophenol tartaric acid UNKNOWN 0.066 4.148
Export results
Show all
Select
#No
API
Coformer
Status
H_ex[kcal/mol]
H_hb[kcal/mol]
G_mix[kcal/mol]
f_fit
Thu Sep 25 12:19:54 CEST 2014: Cocrystal screening - temperature: 298.15 K. Running time: 13.9000 sec
< Back
Screening No.1
graph No.1
Status FINISHED
Memory 84/3616 MB
Progress 0%

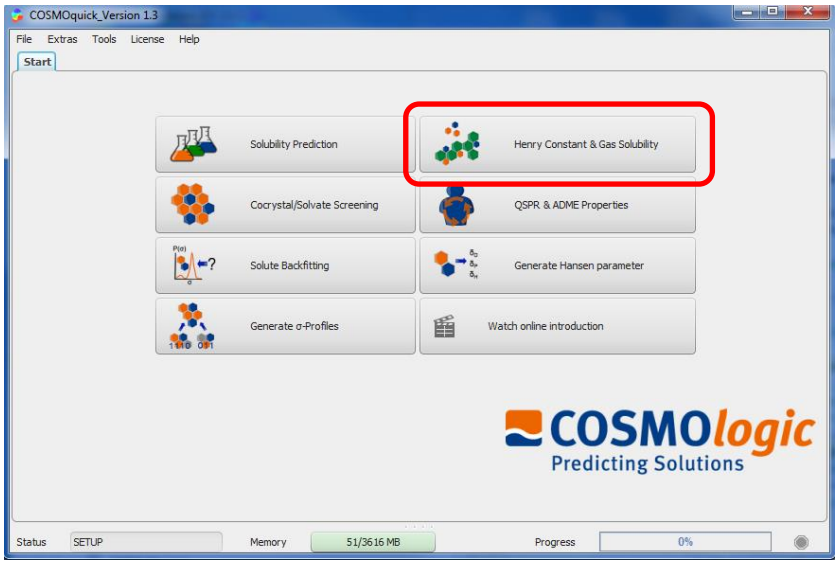
We should find now all pairs which have a low excess enthalpy at the top of the list; those are compounds which have a high probability to form a cocrystal (see also section 3.3). Its also possible to display quantities which describe the part of the enthalpy which is due to hydrogen bonding (H_hb) and the free energy of mixing G_mix of the “cocrystal liquid”. The column denoted “f_fit” contains the results of an empirical screening function which takes into accound the excess enthalpy and the molecular flexibility of the drug and the coformers (see also section 3.3). The trends of those quantities should be the same, but the best ranking is usually obtained by the empirical function f_fit.

Note, that sometimes cocrystal formation is mainly due to an efficient packing in the solid state. Such special cases can not be predicted by the COSMO-RS approach, which relies solely on liquid phase interactions. Furthermore, it can never be ruled out that one of the predicted cocrystals was just missed in the chosen experimental setup. A detailed study of coformer screening with COSMO-RS can be found in reference 5.

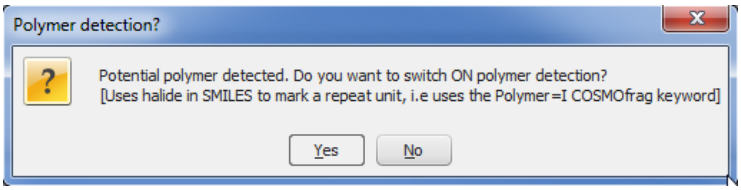
There is a second window available with plots of the computed energies. You may now extract the results either by using copy&paste on the tables (Ctrl+C/Ctr+V) or use the export to excel function.

2.3. Sorption & Solubility in Polymers

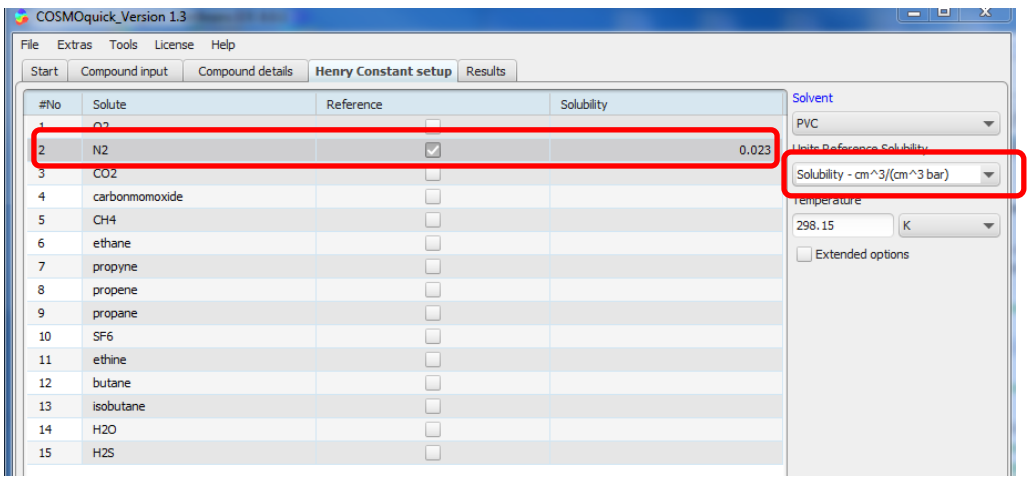
This section explains how to compute the sorption of small molecules from the gas phase into a polymer or any other solvent. This property is usually equivalent to the Henry constant of the molecule within the polymer/solvent system. As a byproduct, the vapor pressure and the solvation free energy are computed. If the solvent is a polymer its repeat unit is decribed by using halide SMILES characters (see section 3.9).



Please select “Henry constant & Gas Solubility” from the first screen. Choose the “import molecules from file” button and load from the exampledata directory the “pvc_sorption.smi”.


Polymer detection?
Potential polymer detected. Do you want to switch ON polymer detection?
[Uses halide in SMILES to mark a repeat unit, i.e uses the Polymer=I COSMOfrag keyword]
Yes
No

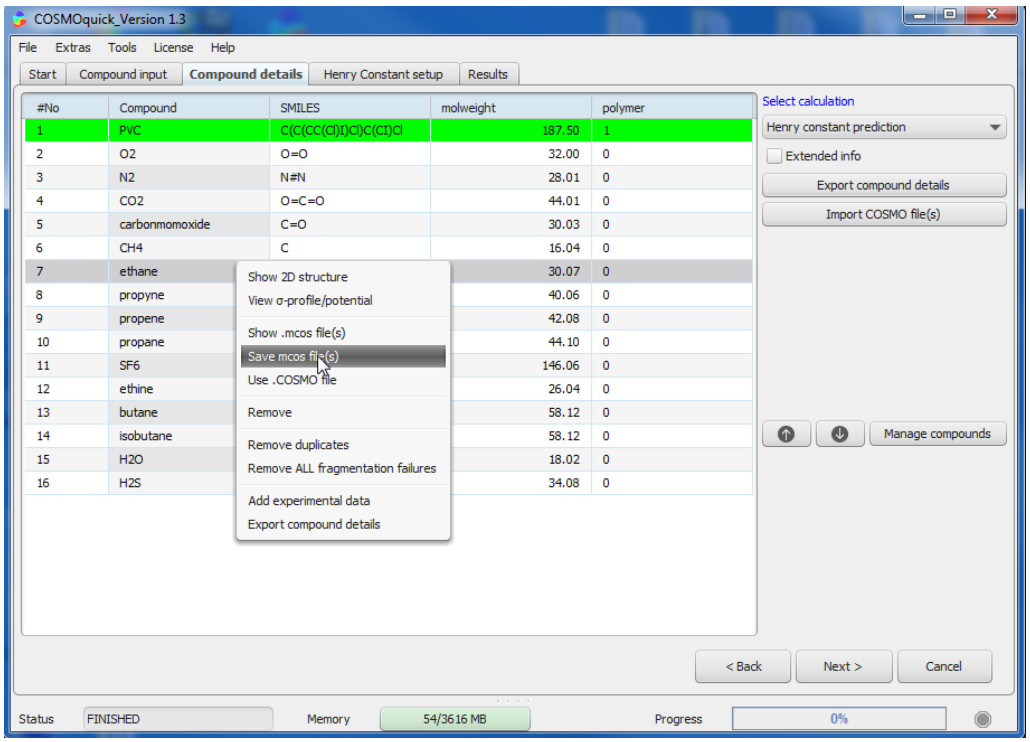
Choose “Yes” to switch on the polymer treatment within COSMO*quick*. For details of the polymer treatment please refer to section 3.9. Now a dataset containing PVC and some small molecules is loaded. If you proceed to the compound details window by clicking “next”, this compound is now labeled as polymer (green colored entry). Continue by choosing screening type “Henry constant”. You should now have a solvent defined (PVC) and see several solutes in the table. If you continue now without further adjustment you would compute the relative solubility constant from the gas phase into the solvent. To get absolute values it is necessary to specify a reference experiment from which a material specific shifting constant for the polymer is computed. In this case we select the solubility of N_2 in PVC as the reference with S = 0.023 $cm^3/(cm^3 bar)$. First we have to select a suitable input from “Units Reference Solubility” the selection box, e.g Solubility in $cm^3/(cm^3 bar)$. Then mark N_2 as reference within the table and type in the solubility.



After starting the calculation via the “Run” button the results are presented in the next window. A polymer shifting constant is computed and correspondingly all solubilities are modified with this shift. Comparison with the experimental data from the Polmyer Handbook (Pauly, S. Polymer Handbook, Permeability and Diffusion Data, Wiley, 2005, 543.) gives a squared correlation coefficient R^2=0.9 for the logarithmic solubility log10(S).

2.4. Exporting .mcos Files

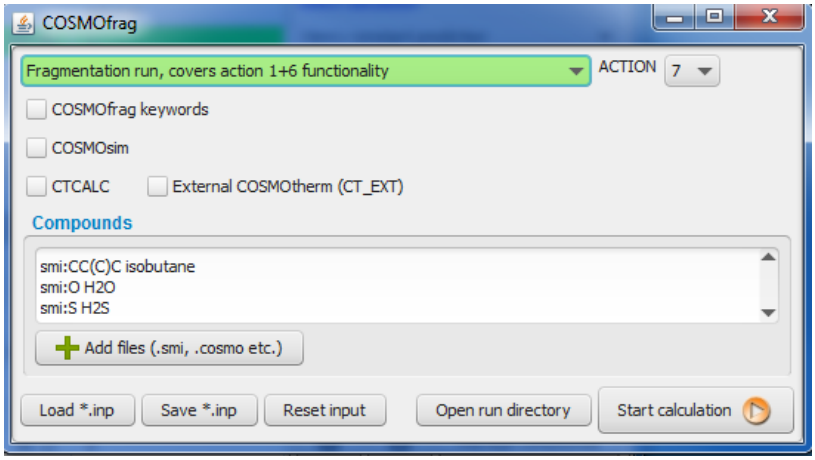
The result of a COSMO*quick* fragmentation calculation for a specific compound is saved in a so-called .mcos file. Those .mcos files contain basically links of all involved fragments which build up the decomposed molecule to their respective compressed .cosmo file (.ccf) within the CFDB. They can be used as any other .cosmo file for subsequent COSMO*therm* calculations. To generate them with COSMO*quick* please activate “Manage compounds” or the context menu within the “Fragment status” panel. Select “Save mcos file” and choose a directory where you want to save the files. There will be a directory “mcos” created, where all the files are saved.



To use them within COSMO*thermX*, you have to use the “File manager” and choose those previously saved .mcos files. PLEASE NOTE: Within COSMO*therm*X a valid path to the COSMOfrag database (CFDB) has to be specified. In “General Settings”, change “Fragment directory (CFDB)” accordingly.

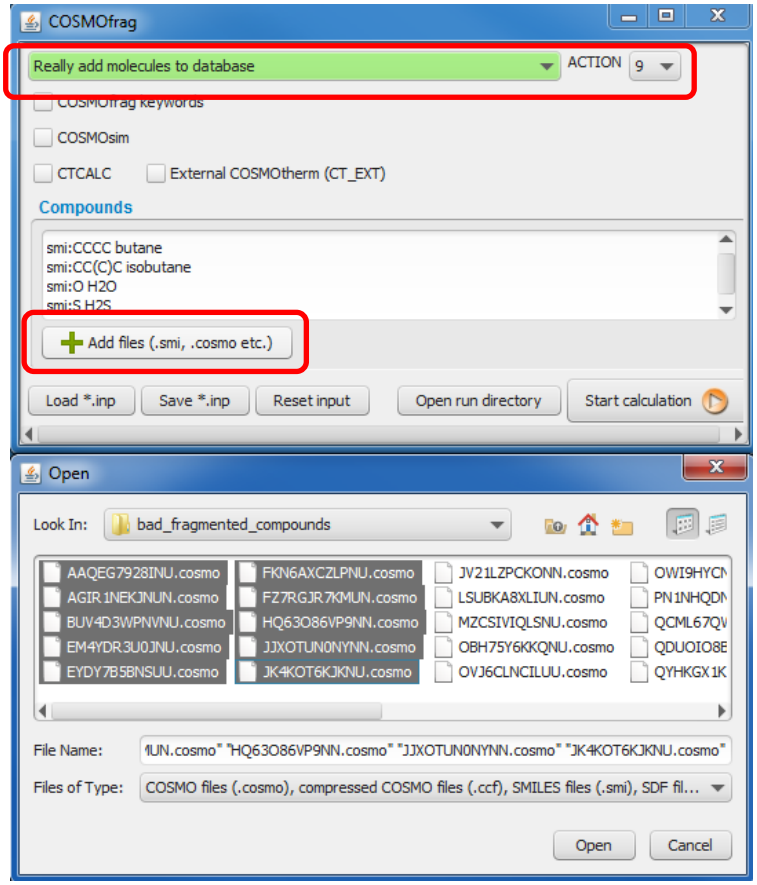
2.5. COSMO*frag* Input Generator

It is now possible with COSMO*quick* to generate input files for COSMO*frag*, which can be submitted from the commandline. This offers some performance advantages and may be useful for highthroughput computations which can not be run and parsed via the graphical user interface. By choosing “Tools->COSMO*frag* calculation” a new window opens with a layout closely resembling the COSMO*frag* command line input:

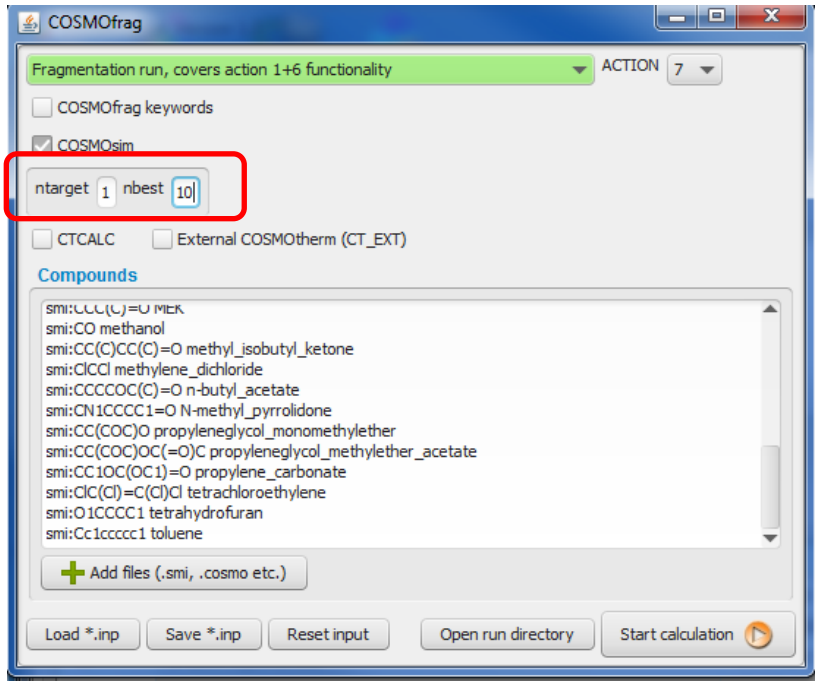


For the details of how to run a COSMO*frag* calculation please consult the manual (Help->COSMO*frag* Reference manual).

Addition of .cosmo files to the database (CFDB): The COSMO*frag* interface may be used to add new molecules to the underlying database. Please note, that you need a quantum chemistry program which is able to create .cosmo files at the SVP level of theory to do this, e.g. TURBOMOLE. Choose "Really add molecules to database" from the pulldown menu and select corresponding cosmo files via "Add files" button. Sometimes it may be useful to choose "Virtually add molecules to database" which leaves the database untouched but gives some information which molecules *would* be added with the current setup. In this respect the MINSADD keyword may be modified which specifies the threshold value of the minimum similarity in a molecule for CFDB addition (default is 2). Values can range from 1 to 7. If you finally press "Start calculation" the molecules in question are added and converted into a compressed format (.ccf), the temporary directory can be accessed via the "Open run directory" in order to look at the COSMO*frag* output.



COSMO*sim* calculations: The COSMO*frag* input generator can also be used to submit molecular similarity calculations based on σ-profiles (COSMO*sim*). Just specify the SMILES or the molecular structures and choose the COSMO*sim* checkbox, where you can define the number of target molecules (ntarget) and the maximal number of closest hits (nbest), please refer also the COSMO*frag* manual for details:



2.6. Other Available Options

There are a few useful tools available for different purposes within COSMO*quick*:

3D structure generation: Once valid SMILES have been created within the compound input panel, they may be converted into 3D structures (.sdf format) using the rdkit (www.rdkit.org). Just select the compounds to be converted via the "Manage compounds" in the Compound input. Please note that those 3D structures should always be checked for correctness.

.fcos file generation: Based on 3D structures (.sdf, .xyz or .COSMO format) COSMO*quick* is able to generate approximate 3D COSMO files. To differentiate from true .cosmo files they have the file suffix .fcos. They may be used for COSMO*sim3D*/COSMO*sar3D* calculations. The .fcos generation option can be found under "Tools". It needs priorily calculated 3D structures and is a stand-alone option.

Additional QSPR descriptos: Additional QSPR descriptors and SMARTS for functional group analysis may be selected at the ADME&QSPR panel. Those descriptors are based on the open source CDK (http://sourceforge.net/apps/mediawiki/cdk/index.php?title=Main_Page, Chemistry Development Kit) software.

3. Technical Details of COSMO*quick*

Currently there are several types of calculations possible with COSMO*quick*. Some of them are COSMO*quick* specific (solubility calculation with several references, cocrystal screening) and some of them can also be carried out with COSMO*frag* at the command line. For those calculations please have a look at the COSMO*frag* manual (e.g. available via the help menu within COSMO*quick*).

3.1. Solubility Calculation

COSMO*quick* is able to use multiple experimental solubilities as reference to refine its solubility prediction. The procedure is outlined below and more details can be found in reference 8. First a number of reference solvents is chosen where we know the solubility e.g. by an experimental measurement. From those n reference solubilites the free energy of fusion $\Delta G_{fus,i}$ is calculated by the following equation (see also reference 4):

$$\Delta G_{fus,i} = \mu_i^{pure} - \mu_i^{solvent} - RT\ln(10)\log 10(x_i)$$

The chemical potentials of the pure liquid solute μ_i^{pure} and the solute in the solvent at infite dilution $\mu_i^{solvent}$ are calculated by COSMO*quick*. The experimental solubility x_i is given as mole fraction in mol/mol. Thus, for every solvent we obtain a free energy of fusion which will be slightly different. Of course, in a perfect model ΔG_{fus} should be the same for any solvent. The basic idea is now to use those differences in the free energy of fusion to correct the chemical potentials within the solvent, where the correction term is adapted to the similarity of the

$$\Delta G_{cor,i} = \Delta G_{fus,i} - \left\langle \Delta G_{fus} \right\rangle \quad i = 1...n$$

reference solvent and the solvent under scrutiny. Thus, the average free energy of fusion is calculated from the references and a correction term is obtained:

$$\Delta G_{cor,i} = \Delta G_{fus,i} - \left\langle \Delta G_{fus} \right\rangle \quad i = 1...n$$

Then, the sigma potential similarity of each new solvent with each reference is computed and the solvent specific free energy corrections are calculated:

$$\Delta G_{cor,j} = \sum_{i}^{references} w^{A}{}_{ji} \Delta G_{cor,i} \quad j = 1...m$$

The normalized weighting factors w_{ji} are determined by the sigma potential similarity of solvent j and reference i:

$$w_{ij} = \exp\left(- \sum_{m=-0.02}^{m=+0.02} \left| \mu_j(\sigma_m) - \mu_i(\sigma_m) \right| \right)$$

μ_i and μ_j are the sigma potentials of reference j and solvent i, respectively. To avoid the dominance of just one reference the weighting factor is smoothed with an exponent A=0.5 (CQ exponent). Finally, we obtain the solubility for our solute in solvent j by the following equation:

$$x_j = \exp\left[\frac{(\mu_j^{pure} - \mu_j^{solvent} - \Delta G_{cor,j} - < \Delta G_{fus} >)}{RT} \right]$$

Please note that the approach will NOT give back the experimental solubilities for the references themselves. Rather they might get a slightly adapted solubility. COSMO*quick* checks the correction term ΔG_{cor}, if this correction is too large (currently the threshold is 1.5 kcal/mol) the program gives a warning message. This is a strong hint that the corresponding experimental value is inaccurate and should be checked. It is recommened to use a balanced set of reference solvents. For example one could use an unpolar solvent like hexane, a donor-acceptor solvent like water, a pure donor solvent like chloroform and an acceptor solvent like acetone. Thus, the solvent space would be well represented and predictions may become more balanced.

Correction for σ-potentials of alkanes. Currently, solubility trends for a solute in a homologue series of alkanes are not reproduced correctly. To overcome deficiencies of the current COSMO-RS approach concerning the solubility in pure alkanes the following correction for the pseudo-chemical potential is used in COSMO*quick* (only for alkanes):

$$\Delta\mu = \left(f(e)_{qspr} - f(e) \right) E_{dielec} \cdot A$$

A is a constant determined by fitting to experimental data (activity coefficients and solubilities in homologue alkanes) and is determined to A=1.2. E_{dielec} is the dielectric energy of the solute in the virtual conductor of the COSMO approach, $f(e)_{qspr}$ and f(e) are the scaling factors for the dielectric sourrounding. The constant scaling factor of a COSMO*therm* calculation f(e) is corrected with a new scaling factor $f(e)_{qsar}$, which has been adapted to reproduce the behavior of alkanes correctly. This scaling factor is obtained from a QSPR for a set of dielectric constants of

alkanes $f(e)_{qspr} = (\varepsilon_{qspr}-1)/(\varepsilon_{qspr}+0.5)$. The corresponding empirical QSPR equations for linear and branched alkanes are:

$$\varepsilon_{linear} = 2.103 - 0.550 * \exp(-0.157\,n)$$

$$\varepsilon_{branched} = 0.03756 * rb - 0.03011\sqrt{n_{aa} * n_{ag}} - 0.002 * rb^2$$

Where n is the number of alkane C-atoms, rb is the number of ringbonds, n_{aa} the number of alkylatoms and the a_{ag} the number of alkylgroups as given by COSMO*frag*. The regression coefficients for those two equations as compared with experimental data are r^2=0.998 for linear alkanes and r^2=0.96 for the branched alkanes. The final dielectric constant is then obtained via:

$$\varepsilon_{qspr} = \varepsilon_{linear} + \varepsilon_{branched}$$

The regression coefficient for QSPR scaling factor $f(e)_{qsar}$ as compared with the experimentally obtained factor is r^2=0.977. This alkane correction is only used for solubility calculations with reference solvents within COSMO*quick*.

Dissociation correction: In the advanced options menu of a solubility calculation it is also possible to switch on a simple Henderson-Hasselbalch dissociation correction term (Diss. Correct.) for aqueous solutions, which may be used to correct the solubilities of strongly dissociating solutes.

3.2. Solubility Definitions and Unit Conversion

Currently there are many different solubility definitions available in the literature. COSMO*quick* uses the decadic logarithm of the mole fraction (log10(x)) internally for its calculations. To alleviate the conversion between different units a solubility converter can be found under Tools->Solubility converter. The same converter can be found by using the context menu when specifying a mixture/solvent for a solubility run. Currently the following solubility definitions can be used, definitions are according to the ones used in the COSMO*therm* code:

- mole fraction x in [mol/mol]
- decadic logarithm of the mole fraction: log10(x)
- normalized mass fraction c in [g/g]:
  c = x_solute * MW_solute /(x_solute*MW_solute+(1-x_solute)*MW_solvent)
- decadic logarithm of normalized mass fraction: log10(c_solute)
- mass based solubility w in [g/g], definition 2 from COSMO*therm* manual:
  w = x_solute * MW_solute /((1-x_solute)*MW_solvent)
- solubility S in mol/L solution:
  S = x_solute / (V_solute + V_solvent)
- solubility S in g/L solution
  S = x_solute* MW_solute / (V_solute + V_solvent)

3.3. Cocrystal Screening

COSMO*quick* allows for the screening of coformers which may form a cocrystal with a given API. A detailed benchmark study of COSMO-RS predictions for cocrystal formation can be found in reference 5. To compute the likelihood of cocrystal formation we start from a virtually subcooled liquid of the cocrystallization components and neglect the long-range order in the crystal. An

important quantity in this respect is the excess enthalpy H_{ex} (=mixing enthalpy ΔH_{mix}) obtained when mixing the pure component A and B to yield the subcooled cocrystal liquid A_nB_m:

$$H_{ex} = H_{AB} - x_m H_{pure,A} - x_n H_{pure,B}$$

H_{AB} and H_{pure} represent the molar enthalpies in the pure reference state and in the m:n mixture, with mole fractions x_m=m/(m+n) and x_n=n/(m+n). The excess enthalpy H_{ex} of an API and conformer pair gives a good estimate of the propensity to cocrystallize.

Technically, COSMO*quick* performs three calculations to obtain H_{ex}: one for each of the pure components A and B, and one mixture calculation for A and B with the given stoichiometry in the subcooled liquid consisting of the mixture of A and B. Sorting the results according to their excess enthalpies will give a list with those compounds having the highest propensity to cocrystallize at the top.

Based on recent work we have introduced a partial empirical function f_{fit} to improve the results of the cocrystal screening. It takes into account the flexibility of the API and the conformer via the number of rotational bonds (nrot).

$$f_{fit} \sim H_{mix} + a\left(\max(1, nrot_{API}) + \max(1, nrot_{COF})\right)$$

With the constant a=0.5102 which has been determined on a set of about 300 API-coformer pairs from the literature. Highly flexible compounds are thus being punished in a screening. We have not fully understood this effect yet. It is probably of kinetic nature, as more flexible compounds may have a higher barrier for crystallization.

3.4. Solute Backfitting

The aim of this approach is to find a description (i.e. a composed or meta COSMO file) of a compound with a structure that is not well defined like a residue or a polymer, based on its solubility in different solvents. In other words, based on given experimental data a meta COSMO file (so-called .mcos file) is generated via an iterative algorithm which reproduces those experimental data as best as possible. This can subsequently be used to predict other properties, like solubilities in other solvents to find replacements or to predict any other property predictable with COSMO-RS.

The general idea is to create a probe compound consisting of several functional groups or fragment molecules, compute the solubility in M solvents, compare with experimtal data points in those solvent and subsequently adapt the probe compound until a convergency threshold is obtained. In detail the workflow is as follows:

As input M experimental solubilities in M different solvents are needed.

[1] Define N diverse functional groups or molecules and store them in an .mcos file

[2] Get molecular weight, volume and area for all FG solutes *and* all solvents.

[3] Create real weight starting guess vector (row weights) r:

$$r = \{r_1, r_2, ..., r_N\}$$

[4] Compute MW,V and A for the pseudo-solute x according to starting guess r, e.g.

$$V_x = \sum_j^N r_j V_j$$

[5] Compute M combinatorial terms for pseudo solute in each solvent.

[6] Compute one chemical potential of pure pseudo solute x and M chemical potentials of x in all solvents (infinite dilution) and add the combinatorial terms from above.

[7] Convert experimental solubilities into mole fractions using MW or V

[8] Determine squared deviation between expt. solubility and predicted solubility:

$$SSE(r) = \sum_i^M \left[\mu_{self}(r) - \mu_{solv,i}(r) + \Delta G_{fus} - RT \ln(x_i^{\exp})\right]^2$$

[9] Embed 3-8 into optimisation algorithm to update row weights of population and minimize SSE. For the optimization constraints are used keeping the $r_i \geq 0$.
If SSE(r)<threshhold then stop the procedure.

3.5. ADME & QSPR Calculations

The following ADME (Absorption, Distribution, Metabolism, and Excretion) property predictions can currently be carried out with COSMO*quick*:

- **log(S)water**: calculation of the solubility of a molecule in water
- **logKow**: calculation of the Octanol-Water partition coefficient of a molecule
- **logKOC**: calculation of the Organic Carbon (Soil)-Water partition coefficient
- **logBB**: calculation of the Blood-Brain Partitioning coefficient, i.e. the penetration of the blood brain barrier
- **logKHSA**: plasma-protein (Human Serum Albumin) partitioning, i.e. the binding to human serum albumin will be calculated
- l**ogKIA:** calculation of the Intestinal Absorption coefficient

Whereas the water solubility and the logKow are calculated on the basis of COSMO-RS theory, the other coefficients are computed via QSPR equations from so-called σ-moments. This set of descriptors is derived from the σ-profile of a compound and can be used to regress almost any kind of partition property. σ-moments may also be useful descriptors to regress other physico-chemical properties and are printed out in the results tab of those QSPR calculations. For more information on performing ADME calculations with COSMO*frag* please consult reference 1.

In addition to ADME properties a set of physicochemical properties can be computed via QSPR based on COSMO*frag* and COSMO*therm* based descriptors. COSMO*quick* can interpret QSPR models based on a multilinear regression, on a Random Forest model[6] or on gradient boosting models (GBM).[7] Those models can be generated for example by the statistics program suite R and be deployed in the PROP directory. Due to their inherent size tree based model structures like Random Forests or GBMs are saved internally in a compressed format (.rfz or .gbmz) and unzipped into RAM upon use.

- **T(melting).rfz:** An empirical random forest model for the prediction of melting points T_m with an (cross-validated RMSE) accuracy of about 40K.
- **H(fusion).mlr:** A multivariate linear regression model for the enthalpy of fusion ΔH_{fus}. It has a (cross-validated RMSE) accuracy of 2.2 kcal/mol.
- **S(fusion).mlr:** A multivariate linear regression model for the entropy of fusion ΔS_{fus}. It has a (cross-validated RMSE) accuracy of 5.81 cal/(mol K).
- **G(fusion).rf:** A model for the prediction of the free energy of fusion ΔG_{fus} out of the melting point and the enthalpy of fusion with an RMSE=0.8 kcal/mol:

$$\Delta G_{fus} = \Delta H_{fus} - T\frac{\Delta H_{fus}}{T_m}$$

The melting point, ΔH_{fus} and ΔG_{fus} QSPR models may be used for example for the generation of reference data for a solubility calculation. In principle arbitrary QSPRs may be generated and deployed within COSMO*quick*. Linear regression based models can also be created with the help of the QSPR builder (see section 3.6). Please contact COSMO*logic* if you are interested in more details on the generation and deployment within COSMO*quick* of those models.

For the creation of the QSPR models a rich set of descriptor from either COSMO*frag* or COSMO*therm* has been used. Please note that in order to use the variable names with external software packages like R any special characters have been removed. This ensures that the variable names stay unchanged after they have been processed externally. They are shortly summarized in the following (you may also hold the mouse over the variable names within COSMO*quick* in order to obtain those information):

Total_q: Total charge sum from σ-profile

n0.030_e.A2 to X0.030_e.A2: p(σ) ranging from -0.030 e/Å^2 to +0.030 e/Å^2

mu_self: chemical potential oft he pure compound in kcal/mol

h_hb: enthalpy due to hydrogen bonding oft he pure compound in kcal/mol

h_int: internal enthalpy oft he pure compound in kcal/mol

e_dielec: dielectric energy

area: surface area of the molecule in Å^2

M2 – M6: σ-moments

volume: molecular volume in $Å^3$

avratio: ratio of surface area to volume

Macc1 – Macc3: σ-acceptor moments

Mdon1 – Mdon3: σ-donor moments

molweight: molecular weight in g/mol

ringbonds: number of bonds in closed ring

alkylatoms: number of pure carbon atoms belonging to alkylgroups CHx

alkygroups: number of alkylgroups (CHx)_n, separated by none alkylatoms

rotatable_bonds: number of effectively rotatable bonds

internal_hbonds: number of internal hydrogen bonds

conjugated_bonds: number of conjugated bonds

rotbsdmod: general flexibility parameter including rings

tmult: topological multiplicity (“2D symmetry”)

nbr11: linear chain rotational bonds

rbwring: ring flexibility parameter

fragments: number of fragments necessary to compose molecule

frag_quality: The average similarity of atomic spheres as compared to the CFDB database [1:bad 9:perfect hit], see also COSMO*frag* maxstring keyword.

zwitterion_in_water: molecule can form a zwitterion in water 1:true 0:false

sulfoxide – quarternary_mixed_ammonium: number of functional groups as computed by CDK (Chemistry Development Kit)

3.6. QSPR Builder

The QSPR builder module allows for the creation of simple QSPR models based on a multiple linear regression. It may be started from the main menu Tools->Create new QSPR model or via the usual workflow from the compound details tab. It is possible to load semi-colon separated files (.csv) containing any kind of descriptor or one may use COSMO*quick* based descriptors. The latter allows for deployment of those models for laters calls from within the program. Linear

regression models are a linear combination of variables and may look like for example for the enthalpy of fusion:

ΔH_{fus}=-2.85 -1.07*μ_{self} +0.45*h_int + 0.08*M2 + 0.14*Mdon2 -0.13*alkylatoms +0.59*nbr11

Please refer to **Section 3.5** for the meaning of the variables. Models built can be saved and used subsequently for other systems. The linear regression models are evaluated using the root means squared error *RMSE* between predicted and experimental values:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}\left(y_i - f(x_i)\right)^2}$$

Here, N is the number of samples, i.e. molecules, y_i is the experimental property and $f(x_i)$ is the predicted quantity of the model $f(x)$ for a molecule with variables x_i. To avoid the problem of overfitting, the *RMSE* is evaluated within a 5-fold cross-validation. Automatic feature selection can be carried out by a so-called greedy forward selection. Starting with a single variable, the one with the lowest RMSE is selected (within a cross-validation) and added to the model. In the next step the best variable among the remaing ones is selected and added to the model. This is repeated until the RMSE cannot be improved significantly. It is very important that this is done within a cross-validation loop, otherwise feature selection may induce quite severe overfitting leading to useless models. Additionally, variables with zero variance, i.e. which are basically constant and highly correlated variables are discarded automatically. There need to be at least as many molecules as variables for the linear regression in order to have a unique solution for the coefficients of the model.

3.7. Prediction of Hansen Solubility Parameter

Hansen solubility parameters[9] are a useful concept for the characterization of solutes and solvents. They describe the solubility characteristics in terms of 3 parameters δD, δP and δH representing dispersion interaction, permanent dipole-dipole interaction and hydrogen bonding, respectively. The parameters for a new solute are usually determined experimentally by measuring its solubility in a set of different reference solvents with known parameters. COSMO*quick* allows for the estimation of those parameters by carrying out COSMO-RS solubility calculations without the need for an experiment.

The workflow is as follows: First, a solute *x* is defined via its 2-D topology (e.g. by the editor or by directly specifying its SMILES code). Then a COSMO-RS computation of the activity coefficient ln(γ) is carried out on a set of reference solvents. An initial guess is made for the the Hansen parameters δD, δP and δH and an activitiy coefficient for solute *x* in solvent *i* is computed via the equation:

$$\ln \gamma^{x}_{i,Hansen} = \alpha \frac{V_x}{RT}\left[4\left(\delta_x^d - \delta_i^{\,d}\right)^2 + \left(\delta_x^p - \delta_i^{\,p}\right)^2 + \left(\delta_x^h - \delta_i^{\,h}\right)^2\right]$$

The activitiy coefficients as computed via the Hansen distance and COMSO-RS are plugged into a sigmoid equation in order to differentiate between good f(*x*)≈1 and bad solvents f(*x*)≈0.

$$f(x) = \frac{1}{1+\exp\left(\frac{x-a}{b}\right)}$$

Then an optimization procedure varies the Hansen parameters such that the squared difference between those two functions becomes minimal:

$$\sum\left[f(\ln\gamma^{x}_{i,Hansen}) - f(\ln\gamma^{x}_{i,COSMO-RS})\right]^2 = \min$$

The parameters *a* and *b* have been optimized on a grid over a set of 29 reference solvents in order to minimize the Hansen distance between predicted and original values.

3.8. Generation of σ-Profiles /Fragmentation Calculation

A fragmentation is the basis for each subsequent calculation. Instead of carrying out a quantum mechanical calculation to get the σ-surface of a novel compound, COSMO*frag* initiates a look up in the COSMO*frag* database (CFDB) for similar molecules or fragments. The novel molecule is then decomposed into a set of fragments, each of which is represented with its σ-profile within the CFDB. For details of the algorithm please consult reference 1. Thus, an approximated σ-profile of the novel molecule is created, which now may be used as any other COSMO file to carry out COSMO-RS calculations. Additionaly, COSMO*frag* carries out a detailed analysis of the molecules. The fragmentation window contains a lot of useful information which are shown by selecting “Extended info”:

Compound: The name of the compound, which may be changed by selecting the cell.

SMILES: The smiles string of the compound (see section 1.2).

Molweight: The molecular weight in g/mol, which is calculated by COSMO*frag*.

UNIQUECODE: A unique 12 letter code for the compound, as created by COSMO*frag*.

Ringbonds: The number of bonds within rings.

Alkylatoms: The number of alklyatoms of the compound.

Alkylgroups: The number of alkylgroups of the compound.

Rotatable bonds: The number of freely rotable bonds of a molecule. The higher the more flexible the molecule is.

Internal hbonds: Number of potential internal hydrogen bonds.

Conjugated bonds: Number of comjugated bonds

Rotbsdmod: Quantifies the general flexibility including rings.

Tmult: A measure for the topological (2D) symmetry due to identical connectivity.

Nbr11: Rotational bonds of linear chains.

Rbwring: Molecular flexibility due to rings.

Fragments: The number of fragments used to create the approximated σ-profile. Zero fragments means the molecule was just taken out of the CFDB.

Frag_quality: A number in which the average similarity of the atoms as compared to the database (COSMO*frag* "maxstring" variable) is given (0:lowest 9:highest). It can be used to identify those compounds which are possibly not represented reasonably by the compounds currently within the CFDB. From our point of view, a similarity value ≥ 2 can always be regarded as adequate. '0' similarities on the other hand should be replaced in either case. COSMO*frag* therefore denotes these molecules with error code 38.

USMILES: A unique smiles code as generated by COSMO*frag*.

Alkane: The number of C-atoms of a pure alkane. If there are heteroatoms the value is -1. This number is used to apply the alkane correction for solubility calculations (section 3.1).

.cosmo file: The name of the cosmo file used for this compound. Usually this will be a .mcos file as generated by COSMOfrag, but also the location of original .cosmo files may be given here.

Error code: The error code of a COSMOfrag fragmentation run. If error code >0, the fragmentation has failed and then the corresponding row is marked red. Those compounds can not be used for a subsequent property prediction. Please consult the COSMO*frag* manual for an explanation of the error codes. The most common reasons for an error code>0 are that the system is charged, or an invalid SMILES string was given in the input.

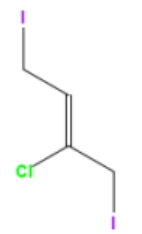
Warn code: The warning code of a COSMO*frag* fragmentation run. If the warning code >0, the corresponding row is marked yellow. Compounds can be used for subsequent property predictions but should be inspected closer. Please consult the COSMO*frag* manual for an explanation of the warning codes.

Polymer: Gives a 1 for a molecule which has been fragmented according to the POLYMER=X options of COSMO*frag* and a 0 for normal molecules.

Charge: Gives the formal integer charge of a molecule as taken from the SMILES.

3.9. Treatment of Polymers

Because there exists no official encoding of polymers as SMILES, COSMO*quick* uses a workaround to mark a polymer repeat unit. Head and tails of a monomer are labeled with the SMILES character usual reserved for halides, for example for polychloroprene the corresponding SMILES is: "C(=C(Cl)Cl)Cl".



Polychloropren

In this case head and tail of the repeat unit are marked by Iodine, but F,Cl or Br are also possible. The molecule is treated internally as infinite cyclic chain, and no molecular weight effects or structural effects are taken into account. COSMO*quick* automatically detects if there are SMILES which have an even number of "I" characters. Alternatively, different halides can be choosen within the global options menu. For very small repeat units it is recommended to

define a dimer or trimer for a more balanced σ-profile composition.

For calculations involving COSMO-RS properties the combinatorial contribution to chemical potential should be switched OFF, i.e. use “Treat solvent as a polymer“ option for Henry constants or polymer solubilities.

3.10. Treatment of Charged Molecules

The COSMO*quick* database (CFDB) contains meanwhile the most common charged functional groups and therefore most charged molecules and zwitterions can be used. This may be useful for example for the creation of .fcos files (approximated .cosmo files from 3D structures) for a subsequent COSMO*sim3D* or COSMO*sar3D* calculation. However, we do not recommend currently using charged species for property prediction. If you try to use a charged molecule for such a task, this will give a warning message, which has to be switched off in the global options menu.

3.11. Scripting in COSMO*quick*

A still somewhat experimental feature is the use of scripting to access internal COSMO*quick* routines. COSMOfrag itself can be scripted at the command line, but in some cases in may be useful to apply the specific workflows which are implemented in COSMO*quick*. Because COSMOquick is JAVA-based a natural choice for scripting access is the Python implementation Jython (http://www.jython.org/). Jython is a fully functional JAVA-based Python implementation and allows for access of any JAVA libraries. The following code gives an example on how to screen on several solutes with Jython and COSMO*quick*:

```
'''
Jython based solubility screening script using COSMOquick libraries
Computes solubility of drugs in different solvents
@author: Christoph Loschen
@copyright: COSMOlogic GmbH 6 Co.KG
'''

import sys
sys.path.append("/home/loschen/COSMOlogic/COSMOquick14/COSMOquick/COSMOquick.jar")
sys.path.append("/home/loschen/COSMOlogic/COSMOquick14/extlib/COSMObasics.jar")
sys.path.append("/home/loschen/COSMOlogic/COSMOquick14/extapps/JChempaint/cdk-1.4.18.jar")
sys.path.append("/home/loschen/COSMOlogic/COSMOquick14/extlib/jfreechart-1.0.17.jar")

from de.cosmologic.cosmoquick.model import CQInterface
from de.cosmologic.cosmoquick.model import CQModel


if __name__ == '__main__':
    #read settings file, can be modified via GUI
    CQInterface.readSettings("/home/loschen/COSMOlogicAppData/COSMOquick14/config/settings.xml")
    exampledir="/home/loschen/COSMOlogicAppData/COSMOquick14/exampledata"

    #list of solutes for screening
    soluteList=["N(C(=O)C)C1=CC=C(O)C=C1 paracetamol","N(C(=O)C)C1=CC=C(O)C=C1
sulfadiazine","C1=NC(=C([NH]1)C)CSCCNC(NC#N)=NC cimetidine"]

    #read solvents from file
    f = open(exampledir+"/solvents.smi", "r")
    solvents = f.read()
    f.close

    solList=[]
    nameList=[]
    #switch on QSPR for GFusion computation
    CQInterface.useGfusionQSPR(True)
```

```
for solute in soluteList:
    #combine solute + solvents
    molset=solute+"\n"+solvents
    #print molset
    cqModel = CQModel()
    cqModel.startFragmentation(molset,False)
    #cqModel.printFragmentationInfo()
    cqModel.setupSolubScreening()
    cqModel.startRefSolubCalculation()

    #collection of results
    for i,m in enumerate(cqModel.getMixtures()):
        if i==0:
            #solute
            solutename=m.getLabel()
            continue
        solList.append(m.getSol_g_p_l())
        nameList.append(solutename+" in "+m.getLabel())

for name,solubility in zip(nameList,solList):
    print "%-64s: %4.2f" %(name,solubility)
```

The script iterates over 3 solutes and computes the solubility in a set of different solvents using a QSPR for the free energy of fusion.

Perequisites for such a scripting are:

- Installation of COSMO*quick* GUI in order to get a settings.xml file with actual paths and directories
- Download of the recent jython version (e.g. 2.7 from sourceforge)
- Adapt paths for .jar archives locations and settings.xml in the jython script (use sys.path.append command as indicated in the example script or set the java CLASSPATH environment variable)

Call jython script with java call: e.g. ~/COSMOlogic/COSMOquick14/jre/bin/java -jar jython-standalone-2.7-b3.jar screening.py

References

1. Hornig, M. & Klamt, A. COSMO*frag*: A Novel Tool for High-Throughput ADME Property Prediction and Similarity Screening Based on Quantum Chemistry *J Chem Inf Model,* **2005***, 45*, 1169-1177.
2. Eckert, F. & Klamt, A. Fast solvent screening via quantum chemistry: COSMO-RS approach *AIChE J* **2002***, 48*, 369-385.
3. Klamt, A. The COSMO and COSMO-RS solvation models, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, 1, 699-709.
4. Klamt, A.; Eckert, F.; Hornig, M.; Beck, M. E. & Bürger, T. Prediction of aqueous solubility of drugs and pesticides with COSMO-RS, *J Comput Chem,* **2002***, 23*, 275-281.
5. Abramov, Y.A.; Loschen, C.; Klamt, A. Rational coformer or solvent selection for pharmaceutical cocrystallization or desolvation*, J. Pharm. Sci.* **2012**, *101*, 3687.
6. Breiman, L. Random Forests, Machine Learning **2001**, *45*, 5.
7. Freund Y., Schapire R.E., A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting, *Journal of Computer and System Sciences* **1997**, *55*, 119.
8. Loschen, C. & Klamt, A. COSMO*quick*: A Novel Interface for Fast σ-Profile Composition and Its Application to COSMO-RS Solvent Screening Using Multiple Reference Solvents, *Ind. & Eng. Chem. Res.* **2012***, 51*, 14303.
9. Hansen, C.M., The three dimensional solubility parameter – key to paint component affinities I. – Solvents, plasticizers, polymers and resins, *J. Paint Technol.* **1967**, 39, 104.

Index

.fcos file 20
.mcos Files 17
3D structure 20
ACC, acceptor 11
add new molecules 18
Addition of .cosmo files 18
ADME 24
alkane correction 22, 29
Alkygroups 26
Alkylatoms 26
area 25
Avratio 26
BP-SVP-COSMO level 3
CDK 26
CDK software 20
CFDB 1, 2, 18
Charged Molecules 30
CIR – chemical resolver identifier 5
Cocrystal Screening 12, 22
compound setup 8
Conjugated_bonds 26
Correction for σ-potentials of alkanes 21
COSMO*frag* 2, 3
COSMO*frag* database (CFDB) 7
COSMO*frag* executable 5
COSMOquick.vmoptions 2
COSMO-RS theory 1
COSMO*sim* 5, 19
CQ exponent 21
D-A, donor-acceptor 11
database 18
descriptor 25
Diss. Correct. 22
Dissociation correction 22
DON, donor 11
e_dielec 25
energy of fusion 20
Error code 29
excess enthalpy 15, 23
f_{fit} 23
Frag_quality 26
Fragmentation Calculation 28
Fragments 26, 29
G_{fit} 23
h_hb 25
H_hb 15
h_int 25
Hansen Parameter 27
Hex 15
  excess enthalpy 15
hydrogen bonding 15
InChi 2
Internal_hbonds 26
Jython 30
License 3
limitations 2
LOAD 5
log window 5
log(S)water 24
logBB 24
logKHSA 24
logKIA 24
logKOC 24
logKow 24
M2 25
Macc1 26
Manage compounds 9
maxstring 26
Maxstring 29
Mdon1 26
Molweight 26
mu_self 25
Nbr11 26
NIH web-service 2
NONP, nonpolar 11
Polymers 29
Proxy-Server 2
QSPR 20, 24
QSPR Builder 26
QSPR descriptos 20
QUICKLOAD 5
Rbwring 26
rdkit 20
reference solubilities 7
Ringbonds 26
Rotatable bonds 28
Rotatable_bonds 26
Rotbsdmod 26
Save mcos file 17
Scripting 30
sigma potential similarity 21
SMARTS 20
SMILES 2, 13, 28
Solubility Calculation 20
SOLUBILITY CONVERTER 5
Solute Backfitting 23
solvation free energy 15
Solvent Screening 7
Sulfoxide 26
Tmult 26
Total_q 25
UNIQUECODE 28
USMILES 29
vapor pressure 15
Volume 26
Warn code 29
Zwitterion_in_water 26
σ-profile/σ-potential 13